

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra elektrotechnologie

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Štefan Mikulics**

Studijní program: Elektrotechnika, energetika a management
Obor: Aplikovaná elektrotechnika

Název tématu: **Ovladač sběrnice XpressNet**

Pokyny pro vypracování:

1. Seznamte se s modelem v laboratoři katedry a návrhem úpravy pro řízení prostřednictvím sběrnice XpressNet.
2. Vyberte vhodný mikroprocesorový modul a navrhnete ovladač s využitím tohoto modulu včetně zobrazování stavu ovladače, případně navrhnete rozšíření modulu o potřebná ovládací tlačítka, respektive displej.
3. Navrhnete/upravte potřebné programové vybavení.
4. Ověřte funkčnost řešení.

Seznam odborné literatury:

- [1] Kuřík, O.: Studie řízení technologického systému, diplomová práce, ČVUT v Praze, FEL, Praha 2011
- [2] Perner, J.: Řízení a vizualizace technologického systému, diplomová práce, ČVUT v Praze, FEL, Praha 2005
- [3] XpressNet Specification, Lenz electronic GMBH,
<http://www.lenzusa.com/1newsite1/Manuals/xpressnet.pdf>

Vedoucí: Ing. Karel Künzel, CSc.

Platnost zadání: do konce letního semestru 2016/2017

Ing. Karel Dušek, Ph.D.
vedoucí katedry



prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 8. 3. 2016

České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra elektrotechnologie



Bakalářská práce

Ovladač sběrnice XpressNet

Štefan Mikulics

Vedoucí práce: Ing. Karel Künzel, CSc.

Studijní program: Elektrotechnika, energetika a management, Bakalářský

Obor: Aplikovaná elektrotechnika

23. května 2016

Poděkování

Rád bych poděkoval svému vedoucímu Ing. Karlu Künzelovi, CSc. za jeho cenné rady a odborné vedení při zpracovávání této bakalářské práce.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Soběslavi dne 23. 5. 2016

.....

Abstract

The Bachelor thesis deals with design and implementation of a XpressNet driver that allows control model train locomotives and other equipment on a railway.

The thesis explains XpressNet bus and serial communication UART. It also describes selected hardware and an analyser that was made to debug the driver. In the conclusion was verified functionality of the devices.

Key words

XpressNet bus driver, serial communication, UART, railway model

Abstrakt

Bakalářská práce se zabývá návrhem a realizací ovladače sběrnice XpressNet, který umožňuje komplexní ovládání modelů lokomotiv a dalších zařízení na kolejišti.

Součástí práce je vysvětlení principu sběrnice XpressNet a sériové komunikace UART. Dále popisuje vybraný hardware a analyzátor, který byl vyroben k ladění ovladače. V závěru práce byla ověřena funkčnost daných zařízení.

Klíčová slova

Ovladač sběrnice XpressNet, sériová komunikace, UART, model kolejiště

Obsah

1	Úvod	1
1.1	Cíl bakalářské práce	1
1.2	Model kolejiště	1
1.2.1	Současný stav modelu kolejiště	1
1.2.2	Senzorový systém kolejiště	3
1.3	Řízení modelu kolejiště	3
2	XpressNet	5
2.1	Architektura	5
2.2	UART	5
2.2.1	Asynchronní přenos	6
2.2.2	Paritní bit	6
2.3	Formát komunikace	7
2.3.1	Call Byte	7
2.3.2	Header Byte	8
2.3.3	Data Byte	8
2.3.4	Error Byte	8
2.4	Časování XpressNet	8
2.5	Připojení XpressNet zařízení	8
2.6	Nevyžádané zprávy	9
2.7	Vlastnosti XpressNet zařízení	9
2.8	Vlastnosti XpressNet příkazové stanice	10
3	Použitý hardware	11
3.1	Vývojový kit STM	11
3.1.1	STM32F429 - DISC0	11
3.1.2	STM32F429 - DISC1	12
3.2	Rotační spínač P-RE24	13
3.3	Plošný spoj	13
4	Vysílané pakety	15
4.1	Pakety vysílané ovladačem	15
4.1.1	Změna rychlosti a směru lokomotivy	15
4.1.2	Žádost o obnovení normálního provozu	16
4.1.3	Potvrzení reakce	16

4.2	Pakety vysílané analyzátořem	17
4.2.1	Příkazové okno	17
4.2.2	Normální provoz obnoven	17
4.2.3	Nouzové zastavení	17
4.2.4	Vypnuté napájení sítě	17
4.2.5	Chybový přenos	18
4.2.6	Neznámá instrukce	18
5	Realizace ovladače lokomotivy	19
5.1	Popis ovládání ovladače lokomotivy	19
5.2	Program ovladače lokomotivy	20
5.2.1	Hlavní program main.c	20
5.2.2	Knihovna uart.c	20
5.2.3	Knihovna rotary_encoder.c	21
5.2.4	Knihovna communication.c	21
5.2.5	Knihovna gui.c	22
5.2.6	Knihovna touch.c	22
6	Realizace analyzátořu	25
6.1	Popis ovládání analyzátořu	25
6.2	Popis programu analyzátořu	26
6.2.1	Hlavní program main.c	26
6.2.2	Knihovna configuration.c	26
6.2.3	Knihovna communication.c	27
6.2.4	Knihovna gui.c	27
6.2.5	Knihovna touch.c	28
7	Ověřeni komunikace	29
7.1	Vysílání pro testování časování	29
7.2	Vysílání „Změna rychlosti a směru lokomotivy“	29
7.3	Vysílání „Nouzové zastavení“ a „Obnovení nor. provozu“	30
7.4	Vysílání „Vypnuté napájení sítě“	31
7.5	Vysílání „Neznámá instrukce“	31
7.6	Vysílání s chybou v přenosu	31
7.7	Komunikace reálného XpressNet systému	32
8	Zhodnocení práce	33
8.1	Ovladač lokomotivy	33
8.2	Analyzátoř XpressNet	33
8.3	Plošný spoj	33
8.4	Porovnání s reálným XpressNet systémem	33
8.5	Celkové zhodnocení	34
A	Schéma zapojení	37

B	Deska plošného spoje	39
B.1	Přední strana - měď	39
B.2	Zadní strana - měď	40
B.3	Vrty	40
C	Obsah přiloženého CD	41
C.1	„Dokumentace“	41
C.2	„Plosny spoj“	41
C.3	„XpressNetAnalyser“	41
C.4	„XpressNetDriver“	41

Seznam obrázků

1.1	Model kolejiště [2]	2
1.2	Schéma kolejiště	2
2.1	Asynchronní přenos	6
2.2	Posloupnost paketu	7
2.3	Konektory RJ-12 a DIN [3]	9
3.1	STM32F429 - DISCO [7]	12
3.2	Rotační spínač P-RE24 [1]	13
3.3	Plošný spoj osazený vývojovými deskami	14
5.1	Ovladač - Hlavní nabídka a numerická klávesnice	19
5.2	Ovladač - Nastavení	20
6.1	Ovladač - Hlavní nabídka a numerická klávesnice	25
7.1	Sejmutý průběh - Odezva ovladače	29
7.2	Sejmutý průběh - Změna rychlosti a směru lokomotivy	30
7.3	Sejmutý průběh - Nouzové zastavení	30
7.4	Sejmutý průběh - Obnovení provozu	30
7.5	Sejmutý průběh - Vypnutí napájení sítě	31
7.6	Sejmutý průběh - Neznámá instrukce	31
7.7	Sejmutý průběh - Chybový přenos 1	31
7.8	Sejmutý průběh - Chybový přenos 2	32
7.9	Sejmutý průběh - Změna rychlosti a směru lokomotivy	32
7.10	Sejmutý průběh - Změna rychlosti a směru lokomotivy	32

Seznam tabulek

2.1	Příklad sudé a liché parity	6
2.2	Význam 5. a 6. bitu [3]	7
2.3	Zapojení RJ-12 a DIN [3]	9
4.1	Změna rychlosti a směru pro 14 rychlostních kroků [3]	15
4.2	Změna rychlosti a směru pro 27 rychlostních kroků [3]	15
4.3	Změna rychlosti a směru pro 28 rychlostních kroků [3]	16
4.4	Změna rychlosti a směru pro 128 rychlostních kroků [3]	16
4.5	Žádost o obnovení normálního provozu [3]	16
4.6	Potvrzení reakce [3]	16
4.7	Příkazové okno [3]	17
4.8	Normální provoz obnoven [3]	17
4.9	Nouzové zastavení [3]	17
4.10	Vypnuté napájení sítě [3]	18
4.11	Chybový přenos [3]	18
4.12	Neznámá instrukce [3]	18

Kapitola 1

Úvod

Bakalářská práce se zabývá návrhem a realizací ovladače sběrnice XpressNet. Součástí práce je pochopení principu fungování XpressNet protokolu, výběr vhodného hardwaru a napsání programu. V závěru práce bude ověřena funkčnost vyrobeného zařízení.

1.1 Cíl bakalářské práce

Cílem bakalářské práce je navrhnout ovladač sběrnice XpressNet pro model kolejiště. Důvodem vývoje ovladače je plánovaný přechod ze sensorového řízení modelu lokomotivy na řízení pomocí XpressNet příkazové stanice. Pochopením protokolu XpressNet existuje možnost návrhu ovladače, který se může přizpůsobit přímo pro dané kolejiště. Cena oproti komerčním ovladačům je mnohem nižší.

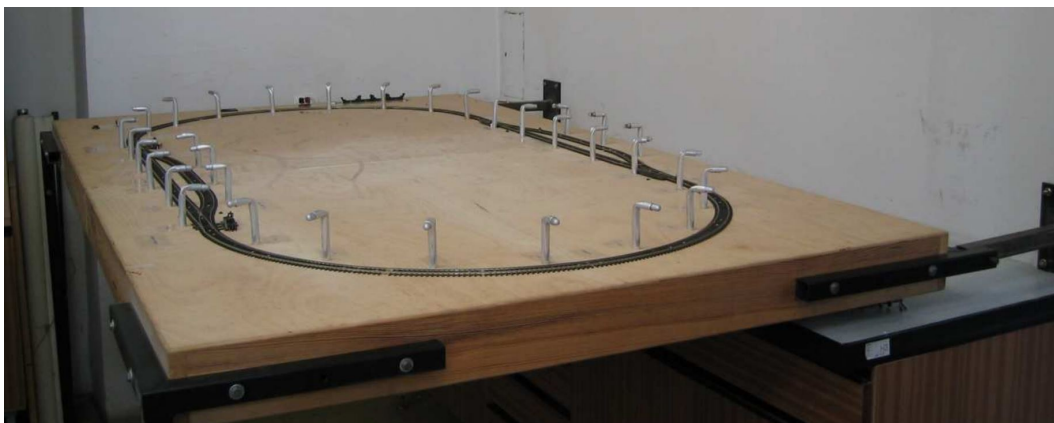
1.2 Model kolejiště

Návrh systému se bude provádět na modelu kolejiště v laboratoři T2:E1-3 katedry elektrotechnologie. S kolejištěm jsem se seznámil v diplomové práci Ondřeje Kuříka, kde bylo kolejiště podrobně popsáno.

Kolejiště bylo postaveno v roce 1988 Janem Štolbou v rámci diplomové práce. Následné modifikace kolejiště nebyly jednotně zdokumentovány. Tento problém vyřešil v roce 2005 Jan Perner ve své diplomové práci, ve které kolejiště podrobně zdokumentoval. V tomto stavu se kolejiště nyní nachází. [2] [4]

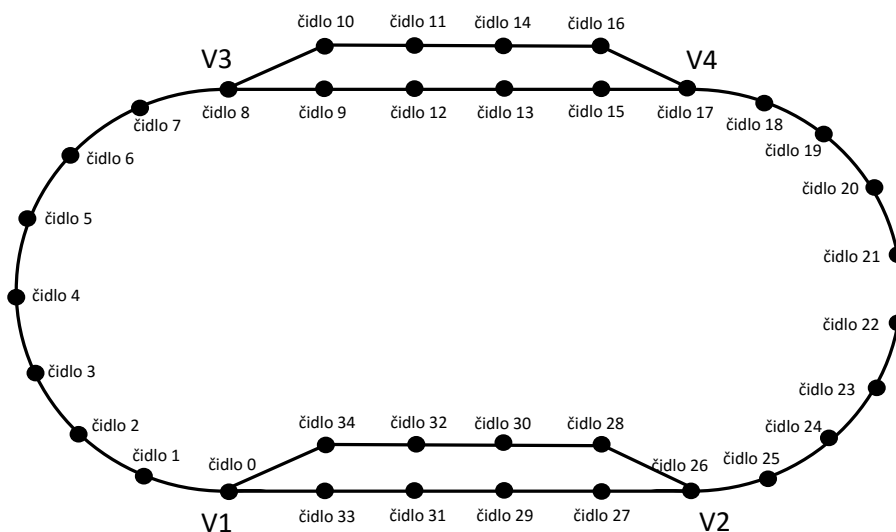
1.2.1 Současný stav modelu kolejiště

Podkladem pro model kolejiště je dřevěná deska o rozměrech (10 x 120 x 200) cm. Kolejiště je ve tvaru oválu se zdvojenými rovnými částmi, kde jsou využity 4 výhybky. Pro vytvoření tohoto tvaru byly využity pouze rovné a polokruhové profily kolejnic. Dále se zde nachází 35 čidel, která patří k sensorovému systému, který bude ještě podrobněji popsán. Schéma kolejiště včetně sensorového systému je zobrazeno na obrázku č. 1.2. Napájení je zajištěno čtyřmi zdroji: [2]



Obrázek 1.1: Model kolejiště [2]

- Napájecí zdroj pro výhybky
- Napájecí zdroj pro osvětlení kolejiště
- Napájecí zdroj 5 V
- Napájecí zdroj 12 V



Obrázek 1.2: Schéma kolejiště

1.2.2 Senzorový systém kolejiště

K sledování okamžité polohy vlaku je využit senzorový systém, který je tvořen soustavou fotosenzorů. Celkový počet fotosenzorů je 35 a je to dáno délkou nejmenší lokomotivy využívané na kolejišti. Vzdálenost mezi nimi je 17,5 cm a díky této vzdálenosti není možné ztratit polohu vlaku. Jejich rozmístění je znázorněno na obrázku č. 1.2. [2]

1.3 Řízení modelu kolejiště

Návrh řízení modelu kolejiště je založen na příkazové stanici XpressNet a ovladači, který bude zrealizován a bude odpovídat standardům XpressNet. Příkazová stanice není součástí modelu kolejiště, a proto při vývoji ovladače bude využíván analyzátor sběrnice, který bude vyroben podle dokumentace XpressNetu. V této bakalářské práci bude navrženo pouze ovládní lokomotiv. Výhybky jsou na kolejišti ovládané pomocí PLC v rámci jiných prací.

Sběrnice XpressNet využívá standartu Direct Command Control (DCC). To znamená, že příkazy, kterými je ovládaná lokomotiva, se vysílají po trati. V modelu kolejiště se nachází jedna příkazová stanice, která je schopna řídit nejen lokomotivy, ale i výhybky, semaforey atd. Příkazová stanice přijímá od ovladačů, kterých může být až 31, požadavek. Ten je následně příkazovou stanicí zakódován a transformován na obdélníkové střídavé napětí. Takto upravený signál přijme dekodér, který je umístěný v lokomotivě, nebo v jiném zařízení na kolejišti. Po přijetí je signál dekodován a zařízení provede danou funkci.

Kapitola 2

XpressNet

XpressNet je síťový komunikační protokol používaný pro řízení modelových železnic. Slouží k propojení vstupních a ovládacích zařízení od společnosti Lenz a pro zařízení, které podporují National Model Railroad Association (NMRA) DCC systémy. NMRA je nezisková organizace, která je zaměřená na oblast modelů železniční dopravy. Dokumentace komunikačního protokolu je veřejně dostupná, a proto je možné připojit jakékoli zařízení, které podporuje tento protokol.

Komunikační protokol je založen na standartu EIA RS-485, který využívá half duplex s jedním párem vodičů. Komunikace tedy probíhá v jeden okamžik pouze jednosměrně. Vlastnosti komunikace pro XpressNet jsou:

- 1 start bit, 8-9 data bytů, 1 stop bit
- Přenosová rychlost 62 500 bit/s

2.1 Architektura

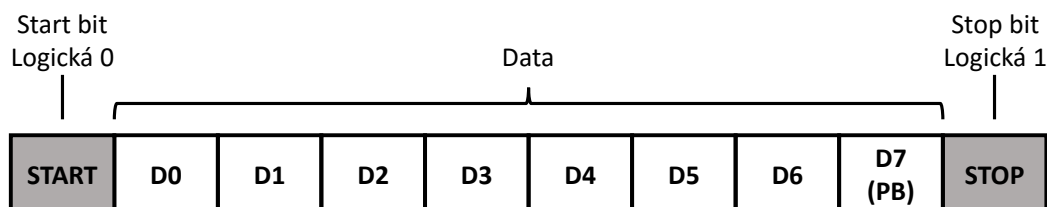
Základem je ovládací zařízení a příkazová stanice. Příkazová stanice přijímá příkazy, pomocí kterých generuje DCC pakety do kolejiště. Dále příkazová stanice spravuje komunikaci po lince, protože po lince lze komunikovat pouze jedním směrem a s jedním zařízením. K tomu slouží tzv. příkazové okno, kterým příkazová stanice dává zařízení povolení pro vysílání jeho požadavku. Když je zařízení obslouženo, tak příkazové okno zaniká a může vzniknout nové pro jiné zařízení. [3]

2.2 UART

Komunikaci mezi deskami bude zajišťovat Universal Asynchronous Receiver/Transmitter (UART) zařízení pro sériovou komunikaci. V případě přímého připojení vývojové desky k XpressNet příkazové stanici by bylo nutné použít převodník úrovní TTL a RS-485.

2.2.1 Asynchronní přenos

V případě asynchronního přenosu není nutné používat hodinový signál, ale je potřeba znát rychlost komunikace mezi zařízeními. Rychlost komunikace v protokolu XpressNetu je 62 500 bit/s. [5]



Obrázek 2.1: Asynchronní přenos

Příklad asynchronního přenosu je na obrázku č. 2.1. Zde musíme rozlišovat zda se jedná o komunikaci s paritním bitem, nebo bez paritního bitu.

V případě komunikace s paritním bitem se přenáší 7bitová data. V klidovém stavu se přenáší logická 1, a proto přenos začíná „Start“ bitem, který má hodnotu logická 0. Poté následují datové bity D0 až D6, které jsou řazeny od nejnižšího bitu k nejvyššímu. Po datech přichází paritní bit (PB) a přenos končí „Stop“ bitem, který má hodnotu logická 1.

Pokud se jedná o komunikaci bez paritního bitu, tak se přenáší 8bitové slovo. Přenos je velmi podobný předchozímu případu. Začíná „Start“ bitem, po kterém následují datové bity D0 až D7 a na konci je „Stop“ bit.

2.2.2 Paritní bit

Paritní bit je bit, který je přidán k přenášeným datům a obsahuje informaci o počtu jedničkových bitů. Využívá se k detekci chyb v přenášených datech. Sudá parita znamená, že počet jedničkových bitů ve slově včetně paritního bitu je sudý. Lichá parita znamená, že počet jedničkových bitů ve slově včetně paritního bitu je lichý. Příklad sudé a liché parity je v tabulce č. 2.1. [6]

7.bitová data	sudá parita	lichá parita
0000000	00000000	10000000
0000001	10000001	00000001
1000001	01000001	11000001
1111001	11111001	01111001

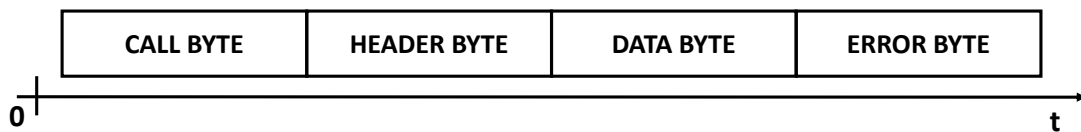
Tabulka 2.1: Příklad sudé a liché parity

2.3 Formát komunikace

Data posílaná mezi ovládacím zařízením a příkazovou stanicí jsou ve formátu tzv. paketu. Tento paket se skládá ze čtyř bytů:

- Call Byte
- Header Byte
- Data Byte
- Error Byte

Protože se jedná o sériovou linku, tak data jsou posílána za sebou. „Call Byte“ je vždy pouze jeden a vysílá jej pouze příkazová stanice. Následující byty vysílá příkazová stanice i XpressNet zařízení. Další v posloupnosti je „Header Byte“, který je také pouze jeden. Následuje 1 až 15 „Data Bytů“ a na konci celé posloupnosti je jeden „Error Byte“. Posloupnost paketu je zobrazena na obrázku č. 2.2. [3]



Obrázek 2.2: Posloupnost paketu

2.3.1 Call Byte

Jedná se o bit, který vyzývá zařízení, aby odeslalo svůj požadavek příkazové stanici. Vysílá jej pouze příkazová stanice. Skládá se z 8 bitů a obsahuje adresu daného zařízení a charakter zprávy. Prvních 5 bitů je vyhrazeno pro adresu, a proto maximální počet připojených XpressNet zařízení je 31. Bity 5 a 6 obsahují charakter zprávy podle tabulky č. 2.2. 7.bit je paritní bit, který slouží pro kontrolu přenosu.

6.bit	5.bit	Význam
0	0	Potvrzení žádosti
0	1	Zpětnovazební vysílání
1	0	Přenosové okno
1	1	Běžná komunikace

Tabulka 2.2: Význam 5. a 6. bitu [3]

2.3.2 Header Byte

Tento byte plní funkci hlavičky před příchodem datových bytů. Dolní 4 bity udávají informaci o počtu následujících datových bytů. Maximální počet je 15. Horní 4 bity obsahují žádost nebo identifikaci, která má být zpracována.

2.3.3 Data Byte

Datový byte se nedá předem definovat, protože záleží na konkrétní komunikaci. Posílá se zde nejčastěji adresa ovládané lokomotivy a požadovaná rychlost.

2.3.4 Error Byte

Jedná se o kontrolní byte, který je počítán funkcí XOR z „Header Byte“ a „Data Byte“. „Call Byte“ se do výpočtu nezahrnuje. Tento byte slouží k ověření vysílané posloupnosti. Při příjmu dat si XpressNet zařízení počítá vlastní kontrolní byte, které po ukončení komunikace porovná s přijatým kontrolním bytem. V případě neshody se paket nebude zpracovávat a odešle na dané zařízení paket „Chyba přenosu“. Stejnou kontrolu provádí také příkazová stanice.

2.4 Časování XpressNet

XpressNet dovede pracovat až s 31 zařízeními. Každé zařízení musí mít adresu v rozsahu 1-31 a příkazová stanice musí všechny tyto zařízení obslužit, ale kvůli sériové komunikaci je možné v jednom okamžiku obsluhovat jen jedno. Příkazová stanice vyšle příkazové okno, kterým se zahájí komunikace s daným zařízením. Po skončení komunikace se zařízením vyšle příkazové okno pro zařízení další v pořadí.

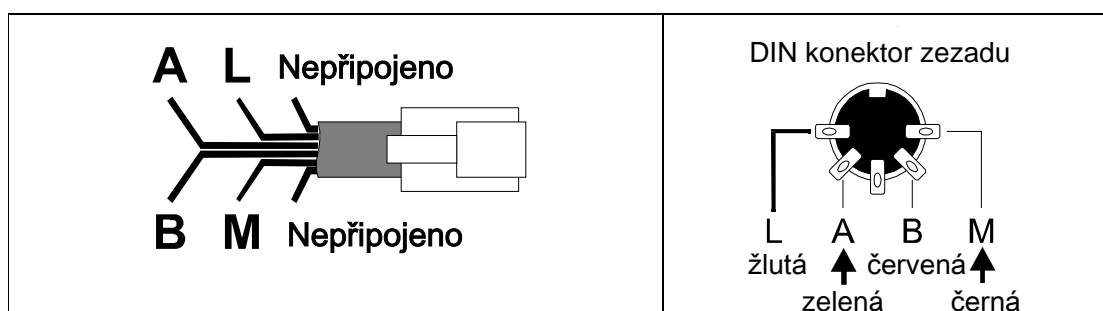
Komunikace je přesně časově definována. Zařízení na síti XpressNet musí být schopno od přijetí příkazového okna odpovědět do 110 μ s. Příkazová stanice musí být schopna přijmout odpověď do 120 μ s od vyslání přenosového okna. V normálních podmínkách musí být zařízení schopno přijmout další přenosové okno v rozmezí 400 až 500 μ s od přijetí předchozího okna. XpressNet zařízení musí být také schopno zpracovávat nevyžádané zprávy, které mohou být přijaty mimo přenosové okno. [3]

2.5 Připojení XpressNet zařízení

Pro připojení se využívá DIN konektor nebo konektor RJ-12. Používají se 4 vodiče a jejich název a funkce jsou:

- L – kladné napájecí napětí pro zařízení
- M – zem
- A – přijímací/vysílací vodič neinvertující
- V – přijímací/vysílací vodič invertující

Zapojení v obou konektorech se zobrazuje na obrázku č. 2.3 a v tabulce č. 2.3. [3]



Obrázek 2.3: Konektory RJ-12 a DIN [3]

Pin	RJ-12	DIN
Pin 1	"C"Kontrolní sběrnice	Nepřipojeno
Pin 2	Zem "M"	a Zem "M"
Pin 3	- RS-485 "B"	- RS-485 "B"
Pin 4	+ RS-485 "A"	+ RS-485 "A"
Pin 5	12 V "L"	12 V "L"
Pin 6	"D"Kontrolní sběrnice	Nepřipojeno

Tabulka 2.3: Zapojení RJ-12 a DIN [3]

2.6 Nevyžádané zprávy

Tyto zprávy musí umět přijímat každé zařízení bez ohledu na příkazové okno. Mezi nevyžádané zprávy patří:

- Obnovení normálního provozu
- Vypnuté napájení tratě
- Nouzové zastavení
- Vstup do servisního módu
- Zpětnovazební vysílání
- Lokomotiva je ovládána jiným zařízením

Prvních pět zpráv je určeno pro všechna zařízení a poslední dvě jsou určeny konkrétnímu zařízení. [3]

2.7 Vlastnosti XpressNet zařízení

Jak již bylo zmíněno XpressNet zařízení musí být schopno mít přesnou časovou posloupnost vysílání, přijímat a vysílat zprávy v paketu a přijímat nevyžádané zprávy. Toto jsou základní podmínky pro každé zařízení připojené do sítě XpressNet. [3]

2.8 Vlastnosti XpressNet příkazové stanice

První vlastnost, kterou musí příkazová stanice splňovat, je vysílání příkazových oken. Po komunikaci se zařízením převádí přijatá data na DCC signál, který se vysílá po trati. Tento signál příkazová stanice vysílá, pokud nepřijme nový příkaz. Například: zařízením se pošle změna rychlosti určité lokomotivě a touto rychlostí se bude pohybovat, dokud rychlost nebude změněna v zařízení. [3]

Kapitola 3

Použitý hardware

3.1 Vývojový kit STM

Pro realizaci ovladače a analyzátoru byly vybrány 2 vývojové desky STM32F429 Discovery. Důvodem výběru této desky bylo, že jsem ji měl již zakoupenou a pracoval jsem na ní. Protože jsem se rozhodl pro dotykové ovládání, tak desky vyhovovaly. Jediné, co bylo potřeba doplnit, byl inkrementální spínač na ovládání rychlosti.

Jelikož desky nebyly zakoupeny ve stejnou dobu, tak se jejich vlastnosti lehce odlišují. Byly použity desky STM32F429 - DISC0 a STM32F429 - DISC1 a jejich vlastnosti budou popsány.

3.1.1 STM32F429 - DISC0

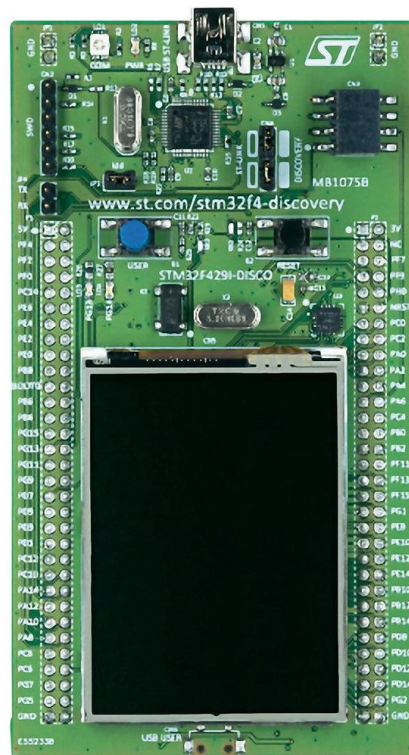
Jedná se o první verzi této desky a její základní vlastnosti jsou: [7]

- ARM mikrokontrolér STM32F429ZIT6 s jádrem Cortex-M4
- Barevný TFT displej s úhlopříčkou 2,4“ s QVGA rozlišením (320 x 240)
- Dotyková vrstva a její řadič
- Externí SDRAM paměť o velikosti 64 Mb
- 2 uživatelské LED, 1 uživatelské tlačítko a tlačítko Reset
- Programátor/debuger ST-LINK V2

Obvod má následující parametry:

- 32bitové jádro s jednotkou pro výpočet pohyblivé řádové čárky (FPU) s maximálním taktem 180 MHz
- 2 MB paměť FLASH, 256 + 4 kB paměť SRAM
- LCD-TFT řadič pro rozlišení až XGA (1024 x 768)

- Napájecí napětí 1,7 V až 3,6 V
- A/D převodník – až 24 kanálů
- 17 časovačů
- 3 sběrnice I2C, 6 sběrnic SPI
- 10/100 ethernet MAC
- Rozhraní pro kameru



Obrázek 3.1: STM32F429 - DISCO [7]

3.1.2 STM32F429 - DISCO1

Tato deska je novější verze předchozí desky. Většina parametrů je stejných, a proto budu zmiňovat pouze ty odlišné. Důvodem zakoupení této desky bylo, že vývojová deska STM32F429 - DISCO již není v žádném obchodě skladem.

Odlišné vlastnosti: [7]

- mbed - systém s možností cloudového připojení
- Virtuální COM port

- Podpora mass storage

Protože na desce byl přidán virtuální COM port, tak na desce nebylo možné využívat U(S)ART1 na pinech PA9 a PA10. Z toho důvodu byl využit UART4 na pinech PC12 a PD2. Na předchozí desce bylo možné využívat U(S)ART1, ale aby zde zůstala možnost záměny desek, tak byl využit také UART4.

3.2 Rotační spínač P-RE24

Pro jednodušší ovládání rychlosti byl k ovladači zakoupen a zprovozněn rotační spínač P-RE24. Spínač je znázorněn na obrázku č. 3.2 a jeho základní vlastnosti jsou: [1]

- 24 impulzů za otáčku
- Proud na kontaktech 1-10 mA
- Životnost 100 000 cyklů

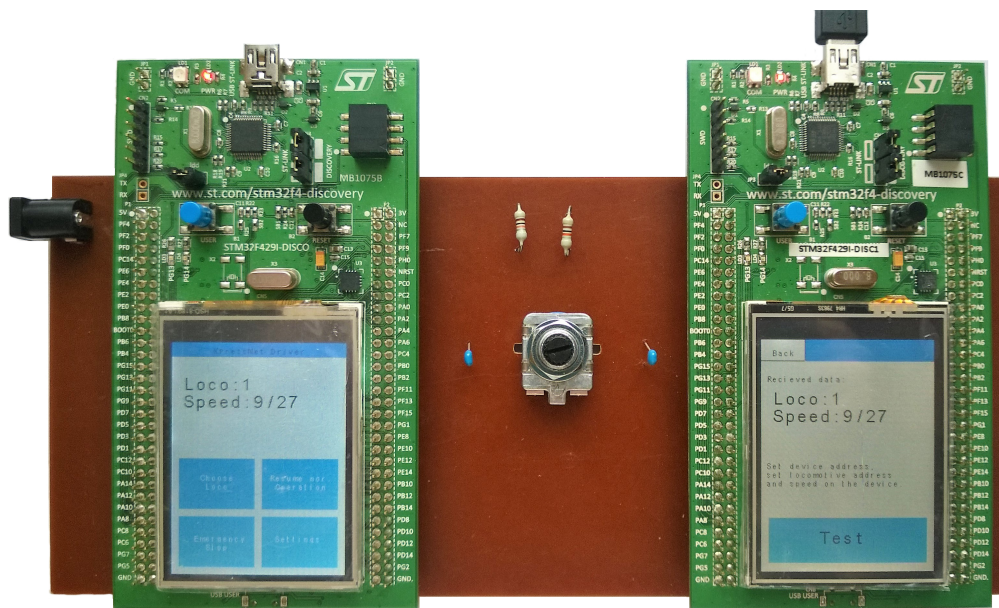


Obrázek 3.2: Rotační spínač P-RE24 [1]

Při použití samotného spínače vznikalo velké množství zákmitů, které vývojová deska zaznamenávala. Řešení tohoto problému mohlo být buď softwarovou, nebo hardwarovou cestou. Já jsem si vybral hardwarové řešení problému, a proto ke spínači byl využit obvod pro omezení zákmitů pomocí paralelně připojených kondenzátoru hodnoty 100 nF. [8]

3.3 Plošný spoj

Při testování propojení desek mezi sebou a připojení inkrementálního spínače bylo využito nepájivé pole, ale protože se kontakty samovolně odpojovaly, tak byl navržen plošný spoj na propojení všech komponentů. Schéma plošného spoje je v příloze A. Plošný spoj byl navržen v programu KiCad.



Obrázek 3.3: Plošný spoj osazený vývojovými deskami

Plošný spoj je osazen 4 dutinkovými lištami, do kterých se připojují vývojové desky. Napravo se připojuje deska s funkcí ovladače a nalevo deska s funkcí analyzátoru. Vzájemná komunikace desek přes piny PC12 a PD2 je zrealizována pomocí 2 vodičů. Mezi deskami se nachází inkrementální spínač P-RE24, který je rozšířen o obvod proti vytváření zákmitů. Spínač je připojen k desce s funkcí ovladače na napětí 3 V a na vstupní piny PD4 a PD5. Plošný spoj je možné napájet externím zdrojem 5 V pomocí napájecího pinu 2,1 mm. Desky je také možné napájet pomocí mini USB na jedné z desek. Na obrázku č. 3.3 je zobrazen plošný spoj osazený vývojovými deskami.

Kapitola 4

Vysílané pakety

Při realizaci ovladače a analyzátoru nebylo využito všech paketů, které se nacházejí v dokumentaci XpressNet. Využité pakety budou stručně popsány a v tabulce zobrazen jejich obsah. V tabulkách P znamená paritní bit, A je 5bitová adresa zařízení a GA je adresa zařízení.

4.1 Pakety vysílané ovladačem

4.1.1 Změna rychlosti a směru lokomotivy

Tento paket zasílá do příkazové stanice informace o změně rychlosti a směru dané lokomotivy. Je zasílán po přijetí příkazového okna a po změně rychlosti na ovladači. Vedle ostatních paketu je zde navíc byte "Identifikace", který slouží k určení rychlostního stupně, který se rozděluje na 14, 27, 28 a 128. Address High a Address Low je dohromady adresa řízené lokomotivy v rozsahu 1 až 9999. Data Byte 3 obsahuje informace o směru - R (0 je směr dopředu a 1 je směr dozadu) a o rychlosti - V. V programu se tento paket nachází pod procedurou *LokomotiveSpeedDirection()*.

	Header B.	Identifikace	Data Byte 1	Data Byte 2	Data Byte 3	Error B.
Bin	1110 0100	0001 0000	AH	AL	R000VVVV	XOR Byte
Hex	0xE4	0x10	AH	AL	RV	XOR Byte
Dec	228	16	AH	AL	RV	XOR Byte

Tabulka 4.1: Změna rychlosti a směru pro 14 rychlostních kroků [3]

	Header B.	Identifikace	Data Byte 1	Data Byte 2	Data Byte 3	Error B.
Bin	1110 0100	0001 0001	AH	AL	R00VVVVV	XOR Byte
Hex	0xE4	0x11	AH	AL	RV	XOR Byte
Dec	228	17	AH	AL	RV	XOR Byte

Tabulka 4.2: Změna rychlosti a směru pro 27 rychlostních kroků [3]

	Header B.	Identifikace	Data Byte 1	Data Byte 2	Data Byte 3	Error B.
Bin	1110 0100	0001 0010	AH	AL	R00VVVVV	XOR Byte
Hex	0xE4	0x12	AH	AL	RV	XOR Byte
Dec	228	18	AH	AL	RV	XOR Byte

Tabulka 4.3: Změna rychlosti a směru pro 28 rychlostních kroků [3]

	Header B.	Identifikace	Data Byte 1	Data Byte 2	Data Byte 3	Error B.
Bin	1110 0100	0001 0000	AH	AL	RVVVVVVV	XOR Byte
Hex	0xE4	0x13	AH	AL	RV	XOR Byte
Dec	228	19	AH	AL	RV	XOR Byte

Tabulka 4.4: Změna rychlosti a směru pro 128 rychlostních kroků [3]

4.1.2 Žádost o obnovení normálního provozu

Pokud se příkazová stanice nachází ve stavu nouzového zastavení, tak tímto paketem se zasílá žádost o normální provoz. Příkazová stanice vyhodnotí, zda lze spustit normální provoz, a případně odešle potvrzení o obnovení normálního provozu. V programu se tento paket nachází pod procedurou *ResumeOperationsRequest()*.

	Header Byte	Data Byte 1	Error Byte
Bin	0010 0001	1000 0001	1010 0000
Hex	0x21	0x81	0xA0
Dec	33	129	160

Tabulka 4.5: Žádost o obnovení normálního provozu [3]

4.1.3 Potvrzení reakce

Tento paket je odesílán v návaznosti na hlášení o chybovém přenosu dat. Pokud ovladač na chybové hlášení nereaguje, tak příkazová stanice nebude dále odesílat příkazová okna tomuto zařízení.

	Header Byte	Error Byte
Bin	0010 0000	0010 0000
Hex	0x20	0x20
Dec	32	32

Tabulka 4.6: Potvrzení reakce [3]

4.2 Pakety vysílané analyzátořem

4.2.1 Příkazové okno

Příkazové okno dovoluje ovladači poslat svůj požadavek na změnu rychlosti a směru lokomotivy. V běžné příkazové stanici se posílá po 150 μ s a mění se adresa zařízení XpressNet. V analyzátořu se adresa předem nastavuje a je konstantní. Příkazové okno je odesíláno pomocí tlačítka.

	Call Byte
Bin	P10A AAAA
Hex	P+0x40+GA
Dec	P+64+GA

Tabulka 4.7: Příkazové okno [3]

4.2.2 Normální provoz obnoven

Tento paket je vysílán, když příkazová stanice přijme žádost o obnovení normálního provozu. V programu se tento paket nachází pod procedurou *NormalOperationResumed()*.

	Call Byte	Header Byte	Data Byte 1	Error Byte
Bin	0110 0000	0110 0001	0000 0001	0110 0000
Hex	0x60	0x61	0x01	0x60
Dec	96	97	1	96

Tabulka 4.8: Normální provoz obnoven [3]

4.2.3 Nouzové zastavení

Tento paket vysílá příkazová stanice, když je potřeba simulovat nouzové zastavení. Paket nelze odeslat, pokud se příkazová stanice nachází v nouzovém stavu. V programu se tento paket nachází pod procedurou *EmergencyStop()*.

	Call Byte	Header Byte	Data Byte 1	Error Byte
Bin	0110 0000	1000 0001	0000 0000	0110 0001
Hex	0x60	0x81	0x00	0x81
Dec	96	129	0	129

Tabulka 4.9: Nouzové zastavení [3]

4.2.4 Vypnuté napájení sítě

Tento paket vysílá příkazová stanice, když je potřeba simulovat vypnuté napájení sítě. V programu se tento paket nachází pod procedurou *TrackPowerOff()*.

	Call Byte	Header Byte	Data Byte 1	Error Byte
Bin	0110 0000	0110 0001	0000 0000	0110 0001
Hex	0x60	0x61	0x00	0x61
Dec	96	97	0	97

Tabulka 4.10: Vypnuté napájení sítě [3]

4.2.5 Chybový přenos

Příkazová stanice při každém příjmu dat počítá Error Byte, který je porovnáván s přijatým Error Byte. Pokud se Error byte od sebe odlišují, je odeslán tento paket. V programu se tento paket nachází pod procedurou *TransferError()*.

	Call Byte	Header Byte	Data Byte 1	Error Byte
Bin	P11A AAAA	0110 0001	1000 0000	1110 0001
Hex	P+0x60+GA	0x61	0x80	0xE1
Dec	P+96+GA	97	128	225

Tabulka 4.11: Chybový přenos [3]

4.2.6 Neznámá instrukce

Tento paket vysílá příkazová stanice, když je potřeba simulovat přijetí paketu, který příkazová stanice neumí zpracovat. V programu se tento paket nachází pod procedurou *InstructionNotSupported()*.

	Call Byte	Header Byte	Data Byte 1	Error Byte
Bin	P11A AAAA	0110 0001	1000 0010	1110 0011
Hex	P+0x60+GA	0x61	0x82	0xE3
Dec	P+96+GA	97	130	227

Tabulka 4.12: Neznámá instrukce [3]

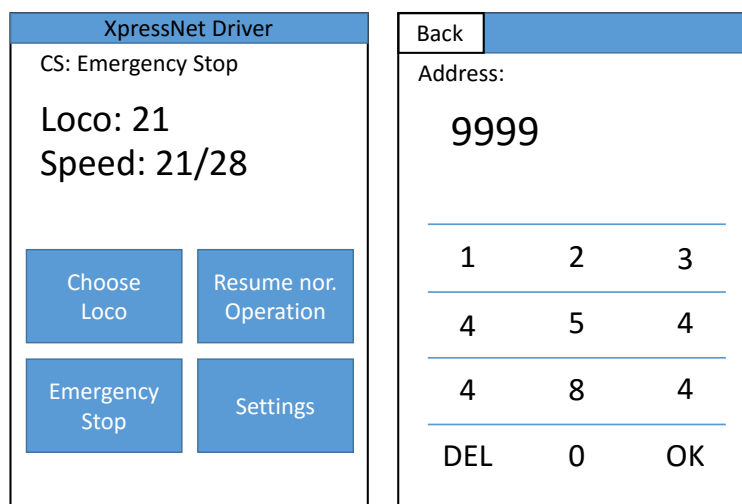
Kapitola 5

Realizace ovladače lokomotivy

Ovladač lokomotivy je naprogramován na desce STM32F429i-DISC0, která byla popsána v kapitole 3. Deska byla navíc rozšířena o inkrementální spínač P-RE24, který se využívá k řízení rychlosti a směru lokomotivy.

5.1 Popis ovládání ovladače lokomotivy

Při zapnutí ovladače se zobrazí hlavní nabídka, na které jsou v horní části displeje informace o ovládané lokomotivě a v dolní části displeje se nacházejí 4 tlačítka. Hlavní nabídka je znázorněna na obrázku č. 5.1 vpravo.

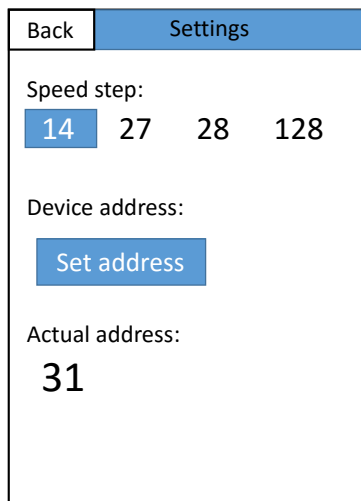


Obrázek 5.1: Ovladač - Hlavní nabídka a numerická klávesnice

Při stisknutí tlačítka „Choose Loco“ se zobrazí numerická klávesnice, kde je možné nastavit adresu lokomotivy v rozsahu 1 až 9999. Adresu potvrdíme tlačítkem OK, nebo smažeme tlačítkem DEL. Numerická klávesnice je znázorněna na obrázku č. 5.1 vlevo.

Tlačítkem „Resume nor. Operation“ se zasílá do příkazové stanice žádost o obnovení normálního provozu. Toto tlačítko je funkční v případě, že se příkazová stanice nachází v nouzovém stavu.

Nastavení ovladače se nachází pod tlačítkem "Settings", kde je možné nastavit rychlostní stupně (Speed step) na hodnoty 14, 27, 28, 128. Dále je zde možné nastavit adresu ovladače pomocí tlačítka "Set Address". Po stisku se zobrazí stejná numerická klávesnice jako v případě "Choose Loco", ale v tomto případě je rozsah adresy 1 až 31. Grafické rozhraní nastavení je znázorněno na obrázku č. 5.2.



Obrázek 5.2: Ovladač - Nastavení

5.2 Program ovladače lokomotivy

Při psaní programu bylo využito vývojové prostředí μ Vision Keil verze 5.0, které bylo vybráno díky velké podpoře vývojových desek od společnosti STMicroelectronics. Při vývoji programu byly využity veřejně dostupné standardní periferní knihovny od STMicroelectronics. Program je rozdělen na několik *.c knihoven z důvodu přehlednosti celého programu.

5.2.1 Hlavní program main.c

Na začátku hlavního programu se postupně volají procedury ke konfiguraci všech potřebných periférií. Poté se program dostane do nekonečné smyčky *while(1)*, kde se volá procedura *TouchControl()*, která stále aktualizuje stav dotykové vstvy. V případě příjmu dat přes UART nastane přerušení, které spustí proceduru *UART5_IRQHandler()*. V případě změny na inkrementálním spínači nastane přerušení, které spustí proceduru *EXTI4_IRQHandler()*.

5.2.2 Knihovna uart.c

Tato knihovna obsahuje pouze jednu rozsáhlou proceduru, která konfiguruje UART.

void UART_Config()

Procedura konfiguruje potřebné parametry ke komunikaci pomocí UART5. Při vysílání dat se využívá pin PC12 a při odesílání dat se využívá pin PD2. Rychlost komunikace je nastavena na 62 500 bit/s. Tato hodnota byla získána z dokumentace XpressNet. Data se vysílají v 8bitovém slově a bez paritního bitu.

5.2.3 Knihovna rotary_encoder.c

Pro inkrementální spínač P-RE24 bylo nutné napsat vlastní knihovnu, která obsahuje proceduru pro konfiguraci spínače a proceduru pro obsluhu spínače.

void RE_Config()

Tato procedura konfiguruje piny a externí přerušení. Inkrementální spínač je připojen na piny PD4 a PD5, které jsou nastaveny jako vstupní piny. Při změně logické hodnoty na pinu PD4 je vyvoláno externí přerušení a spuštěna procedura *RE_Control()*.

void RE_Control()

Tato procedura se spouští při externím přerušení na PD4. Využívá se zde automatu typu Moore, který porovnává předchozí hodnotu a aktuální hodnotu. Podle toho se rozhodne, pokud se proměnná *int speed* má inkrementovat nebo dekrementovat. Proměnná *speed* je omezená proměnnou *int maxspeed*, která se nastavuje v nastavení a jedná se o maximální rychlostní stupeň.

5.2.4 Knihovna communication.c

Tato knihovna je základem celé komunikace, která funguje přes UART5.

void CommunacationControl()

Tato procedura se spouští pomocí UART přerušení, které je vyvoláno při příjmu dat na pinu PD2. Data z registru DR se uloží do proměnné *char ch*. Poté se kontroluje, jestli se jedná o přenosové okno, pokud ano, tak se pošle žádost o změnu rychlosti a směru lokomotivy, pokud ne, tak se proměnná *ch* přidá do pole *unsigned char received_data[10]*, kde se ukládá celá sekvence. Proměnná *char cnt* udává, kolikátý byte byl přijat. Když se *cnt* rovná 1, tedy jedná se o „Header Byte“, zjišťuje se, kolik bude následovat bytů „Data Byte“. Po přijmutí všech se začne počítat „Error Byte“, který se porovná s příchozím „Error Byte“. Aby se spustila procedura *DataAnalyse()*, tak se „Error Byty“ musí rovnat.

void DataAnalyse()

V této proceduře se pomocí podmínek *if* zjistí, zda se jedná o „Obnovení normálního provozu“, „Vypnuté napájení sítě“, „Nouzové zastavení“, nebo „Chybu přenosu“. Datové pakety jsou dány v dokumentaci XpressNet. Po přijetí jedné z těchto zpráv se v horní části displeje zobrazí její název.

unsigned char ErrorByteCalc(unsigned char data[10], char start, char end)

Tato funkce počítá „Error Byte“ pomocí funkce XOR z prvků pole *unsigned char data[10]*. Parametry *char start* a *char end* lze nastavit rozsah zpracovávaných prvků pole. Návrátovou hodnotou funkce je *unsigned char myerrorbyte*.

Procedury k odesílání paketu

Tyto procedury jsou velmi podobné, a proto není nutné popisovat jednotlivě. Posílají se zde datové pakety pomocí UART5. Jedná se o procedury: *Acknowledgement_Response()*, *Resume_Operations_Request()*, *Emergency_Stop()*, *Lokomotive_Speed_Direction()*.

5.2.5 Knihovna gui.c

V této knihovně se vyskytují všechny důležité procedury k uživatelskému grafickému rozhraní.

void LCD_Init_Default()

Tato procedura konfiguruje LCD displej pomocí knihoven od společnosti STMicroelectronics.

Procedury k určitým obrazovkám

GUI_Menu() zobrazuje hlavní nabídku, *GUI_Keyboard()* zobrazuje numerickou klávesnici, která se zobrazuje při nastavování adresy ovládané lokomotivy a adresy ovladače, *GUI_Settings()* zobrazuje nastavení, ve kterém se nachází nastavení rychlostního stupně a adresy ovladače, *GUI_SettingsSpeed()* se stará o aktualizaci rychlosti při ovládaní pomocí inkrementálního spínače, *GUI_Messeges()* zobrazuje informace, které posílá příkazová stanice.

5.2.6 Knihovna touch.c

V této knihovně se vyskytují všechny důležité procedury k obsluze dotykové vrstvy.

void Touch_GUI_Config()

Procedura kontroluje, zda je dotykový displej v pořádku a pokud ano, tak se zobrazí hlavní nabídka a pokud ne, tak se zobrazí chybová hláška „IOE NOT OK. Reset the board and try again.“ na červeném pozadí.

void Touch_Control()

Tato proceduru pomocí podmínky *if* a proměnné *char position* rozlišuje, v jaké části grafického rozhraní se program nachází. Pro každé rozhraní se zde nachází samostatné procedury.

void Touch_KeyboardControl()

Procedura, která zajišťuje funkčnost dotykového ovládání numerické klávesnice.

void Delete_Address()

Procedura, která u numerické klávesnice pomocí tlačítka DEL smaže řádek se zadaným číslem.

Procedury k určitým obrazovkám

V knihovně se také nachází procedury ke každé obrazovce v grafickém rozhraní. Tyto procedury se starají o funkčnost tlačítek na určitých obrazovkách, které byly vyjmenovány v sekci č. [5.2.5](#).

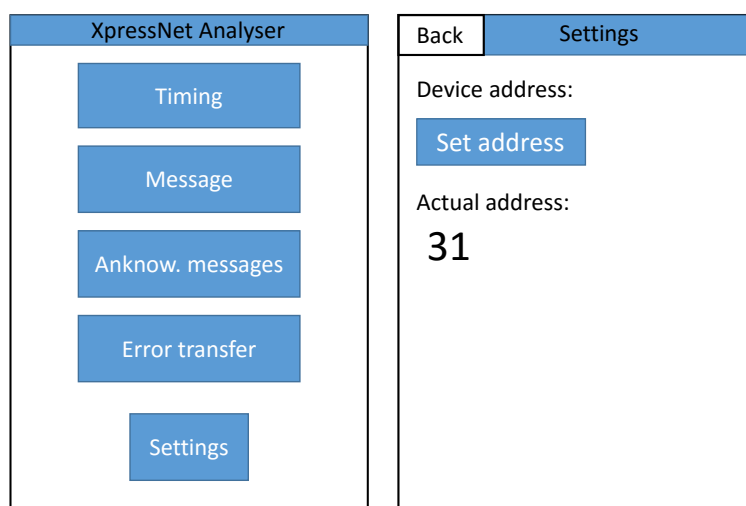
Kapitola 6

Realizace analyzátoru

Pro ověření funkčnosti ovladače byl navržen analyzátor, který je schopen simulovat příkazovou stanici XpressNet. Je zde možnost testování časové odezvy ovladače na příkazové okno, přijímání datových paketů s rychlostí a směrem lokomotivy, odesílání nevyžádaných zpráv do ovladače a simulování chybného přenosu.

6.1 Popis ovládání analyzátoru

Při spuštění analyzátoru se objeví na displeji hlavní nabídka, která obsahuje 4 různé druhy testů a nastavení. Hlavní nabídka je znázorněna na obrázku č. 6.1 vlevo.



Obrázek 6.1: Ovladač - Hlavní nabídka a numerická klávesnice

„Timing“ testuje, zda ovladač je schopen odpovědět na přenosové okno do $110 \mu\text{s}$. Tato hodnota je stanovena v dokumentaci XpressNet a je to jedna z podmínek pro ovladač. Pomocí tlačítka „Test“ začne testování a výsledek se zobrazí na displeji.

„Messege“ testuje, zda ovladač umí odpovědět na přenosové okno změnou rychlostí a směru lokomotivy. Pomocí tlačítka „Test“ se odešle příkazové okno a na displeji se zobrazí přijaté informace.

„Anknow. Messesges“ testuje, jestli ovladač umí přijímat a případně odpovídat na nevyžádané zprávy. Lze odeslat „Vypnutí napájení sítě“, „Neznámý příkaz“ a „Nouzové zastavení“. Pokud ovladač zprávu správně přijme, tak se zpráva zobrazí v horní části displeje ovladače. Po nouzovém zastavení lze na ovladači obnovit normální provoz pomocí tlačítka. Stav provozu se zobrazuje na analyzátoru v dolní části displeje.

„Error Transfer“ testuje, zda ovladač umí reagovat na hlášení o chybě přenosu. Analyzátor pošle přenosové okno, ovladač odešle rychlost a směr lokomotivy. Při výpočtu kontrolního bytu vznikne záměrná chyba, a proto analyzátor odešle chybové hlášení. Výsledek testu se zobrazí na displeji po stisknutí tlačítka „Test“.

Nastavení analyzátoru se nachází pod tlačítkem "Settings". Zde je možné nastavit adresu zařízení, které se bude testovat. Nastavení je znázorněno na obrázku č. 6.1 vpravo.

6.2 Popis programu analyzátoru

Program byl vytvářen stejným způsobem jako program pro ovladač lokomotivy v sekci 5.2. Program je také rozdělen do několika *.c knihoven kvůli lepší přehlednosti.

6.2.1 Hlavní program main.c

Hlavní program se velmi podobá programu ovladače. Na začátku hlavního programu se postupně volají procedury ke konfiguraci všech potřebných periférií. Poté se program dostane do nekonečné smyčky *while(1)*, kde se volá procedura *TouchControl()*, která stále aktualizuje stav dotykové vrstvy. V tomto případě se zde může nastavit pouze přerušování při přijetí dat a spustí se procedura *UART5_IRQHandler()*.

6.2.2 Knihovna configuration.c

Knihovna, která obsahuje funkce pro základní konfiguraci UART5 a TIMER3.

`void Timer_Config()`

Procedura, která konfiguruje TIMER3. Dělička je nastavená na 40, perioda je 500. Přerušování není nutné, protože se k aplikaci používá pouze aktuální hodnota čítače.

`void UART_Config()`

Procedura, která konfiguruje UART5, která využívá porty PC12 a PD2, rychlost je nastavena na 62500 baud. Při přerušování je obsluhována procedura *UART5IRGHandler()*, která resetuje request flag a volá proceduru *CommunicationControl()*. Nastavení UART je stejné jako u ovladače lokomotivy.

6.2.3 Knihovna `communication.c`

Tato knihovna obsahuje důležité funkce pro komunikace pomocí UART5. Strukturou je tato knihovna velmi podobná stejnojmenné knihovně v programu ovladače lokomotivy v sekci 5.2.4, ale procedury jsou odlišné.

unsigned char ParityBit_Calc(unsigned char myChar)

Tato funkce slouží k výpočtu paritního bitu. Paritní bit se počítá z parametru *unsigned char myChar* a je přidán na 7. bit proměnné *char even*. Bity 0 až 6 jsou totožné z *myChar*. Proměnná *even* je návratovou hodnotou funkce.

void CommunicationControl()

Procedura, která slouží k ukládání přijatých dat do proměnné *receiveddata[10]* a jejich částečné analýze. Ve proceduře se také nachází porovnávání přijatého a vypočteného „Error Byte“. Při shodě „Error Bytů“ odkazuje na proceduru *DataAnalyse()*.

void DataAnalyse()

Procedura, která analyzuje pomocí podmínek *if* přijatá data v proměnné *char receiveddata[10]*. Procedura rozezná, zda se jedná o žádost rychlosti a směru lokomotivy, žádost o obnovení normálního provozu, nebo potvrzení žádosti.

unsigned char ErrorByte_Calc(unsigned char data[10], char start, char end)

Tato funkce je převzata z programu ovladače lokomotivy a je popsána v sekci 5.2.4.

Procedury k odesílání paketu

Tyto procedury jsou velmi podobné, a proto není nutné popisovat jednotlivě. Posílají se zde datové pakety pomocí UART5. Jedná se o procedury: *NormalOperationResumed()*, *EmergencyStop()*, *TrackPowerOff()*, *TransferError()*, *InstructionNotSupported()*.

6.2.4 Knihovna `gui.c`

V této knihovně se nachází procedury ke grafickému uživatelskému rozhraní. Funkčnost je velmi podobná stejnojmenné knihovně v programu ovladače lokomotivy. Procedura *LCD_Init_default()* je stejná a byla již popsána v sekci 5.2.5.

Procedury k určitým obrazovkám

Procedury jsou velmi podobné, a proto není nutné je popisovat jednotlivě. Procedury vykreslují danou obrazovku v grafickém rozhraní. Jedná se o procedury: *GUI_Menu()*, *GUI_Transfer()*, *GUI_TransferResult()*, *GUI_Messegas()*, *GUI_Error()*, *GUI_Timing()* a *GUI_Keyboard()*.

6.2.5 Knihovna `touch.c`

V této knihovně se nachází procedury pro funkčnost dotykové vrstvy. Funkčnost je velmi podobná stejnojmenné knihovně v programu ovladače lokomotivy. Procedury `Touch_GUI_Config()` a `Touch_Control()` jsou stejné a byly již popsány v sekci 5.2.6.

Procedury k určitým obrazovkám

Tyto procedury se velmi podobné, a proto není nutné je popisovat jednotlivě. Jedná se o dotykové ovládání na dané obrazovce. Mezi tyto procedury patří: `Touch_Menu()`, `Touch_Transfer()`, `Touch_TransferResult()`, `Touch_Messages()`, `Touch_Error()`, `Touch_Timing()` a `Touch_Keyboard()`.

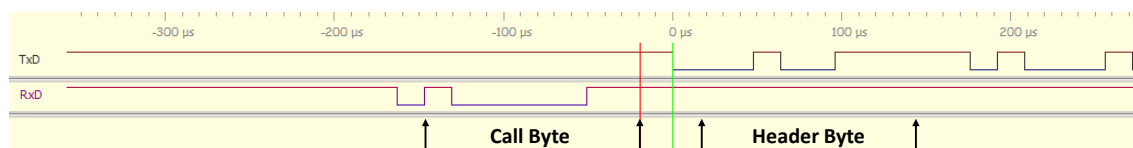
Kapitola 7

Ověření komunikace

Ověření funkčnosti obou zařízení jsem provedl snímáním průběhů logických hodnot na datových vodičích mezi ovladačem a analyzátozem. Snímání probíhalo na zapůjčeném logickém analyzátozem ASIX Omega, který byl propojen s notebookem. Při snímání se využíval program, který je dodáván s analyzátozem. Na zachycení průběhu byla využita aplikace OCam.

Na každém obrázku jsou 2 signály, které jsou popsány následovně. TxD je vysílací datový kabel ovladače lokomotivy a RxD je přijímací datový kabel ovladače lokomotivy.

7.1 Vysílání pro testování časování

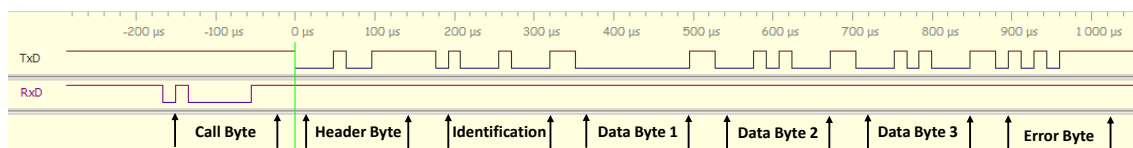


Obrázek 7.1: Sejmутý průběh - Odezva ovladače

Ovladač lokomotivy musí být schopen odpovědět na příkazové okno do 110 μs . Analyzátoz odešle příkazové okno a zároveň se zapne měření času. Po přijetí hlavičkového bytu se měření zastaví a zobrazí se na displeji analyzátozu. Na obrázku č. 7.1 je průběh testování. Vyzývací byte má binární hodnotu 11000001 a hlavičkový byte má 11100100. Odezva je doba mezi červenou a zelenou čarou a v tomto případě je to 19 μs , a proto ovladač v tomto testu vyhověl.

7.2 Vysílání „Změna rychlosti a směru lokomotivy“

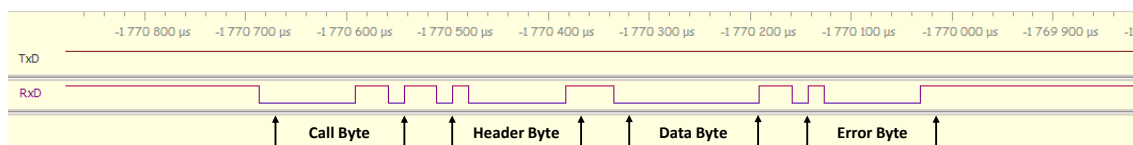
Ovladač lokomotivy přijme příkazové okno a odpoví na to paketem pro změnu rychlosti a směru lokomotivy. Průběh je zobrazen na obrázku č. 7.2.



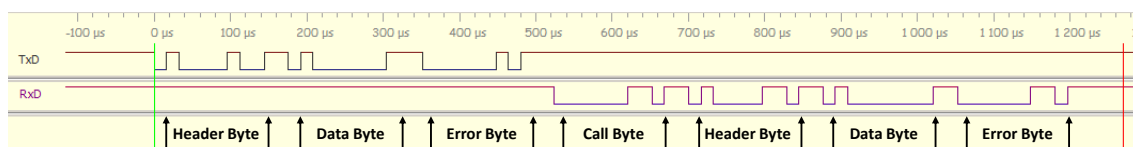
Obrázek 7.2: Sejmутý průběh - Změna rychlosti a směru lokomotivy

„Call Byte“ má hodnotu 11000001 a tedy se jedná o příkazové okno pro zařízení s adresou 1. „Header Byte“ má hodnotu 11100100, a proto budou následovat 4 byty. „Identification“ je 00010001 a to znamená, že rychlostní stupeň je 27. Pod „Data Byte 1“ a „Data Byte 2“ se nachází adresa lokomotivy. „Data Byte 1“ je nulový a „Data Byte 2“ má hodnotu 00010100, a proto adresa lokomotivy je 20. „Data Byte 3“ udává informaci o směru a rychlosti a má hodnotu 00010100, to znamená směr dozadu a rychlost 20. Poslední byte v paketu je kontrolní a má hodnotu 11110101, která odpovídá funkci XOR všech odeslaných bytů.

7.3 Vysílání „Nouzové zastavení“ a „Obnovení nor. provozu“



Obrázek 7.3: Sejmутý průběh - Nouzové zastavení

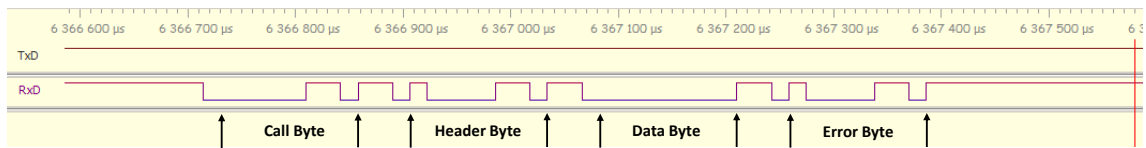


Obrázek 7.4: Sejmутý průběh - Obnovení provozu

V průběhu na obrázku č. 7.3 vysílá analyzátor paket s nouzovým zastavením, který odpovídá paketu „Nouzové zastavení“ v sekci č. 4.2.3. Na tento paket ovladač neodpovídá, ale v horní části se zobrazí zpráva o nouzovém zastavení.

Když se systém nachází ve stavu nouzového zastavení, tak ovladač může odeslat žádost a analyzátor odešle všem zařízením potvrzení. Tato akce je zobrazena na obrázku č. 7.4. Žádost o obnovení normálního provozu byla již popsána v sekci č. 4.1.2 a odpovídá průběhu, který vysílá ovladač. Analyzátor odešle „Potvrzení o obnovení normálního provozu“, které bylo popsáno v sekci 4.2.2 a odpovídá to průběhu, který vysílá analyzátor.

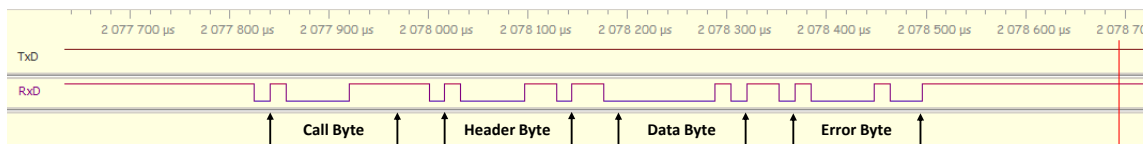
7.4 Vysílání „Vypnuté napájení sítě“



Obrázek 7.5: Sejmutý průběh - Vypnuté napájení sítě

Pomocí tlačítka „Track Off“, které se nachází v „Anknow. Messesges“ se simuluje vypnuté napájení sítě. Po stisku se odešle paket „Vypnuté napajeni sitě“, který byl popsán v sekci č. 4.2.4. Průběh na obrázku č. 7.5 odpovídá již zmíněnému paketu. Ovladač na tento paket neodpovídá, pouze se v horní části displeje zobrazí zpráva.

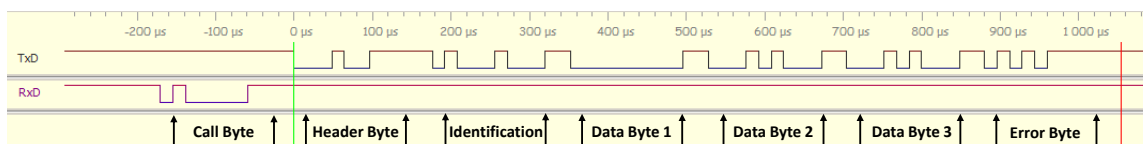
7.5 Vysílání „Neznámá instrukce“



Obrázek 7.6: Sejmutý průběh - Neznámá instrukce

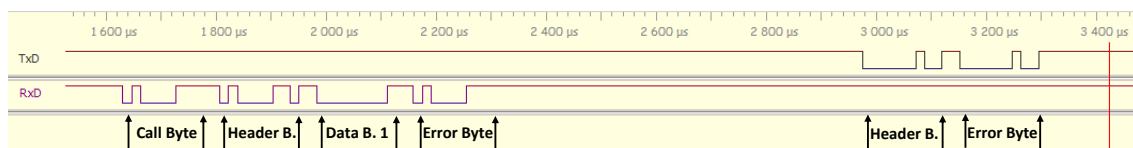
Po stisku tlačítka „Unknown Instruction“, které se nachází v „Anknow. Messesges“ se provede simulace, že příkazová stanice přijme neznámou instrukci a odešle paket, který byl popsán v sekci č. 4.2.6. Průběh na obrázku č. 7.6. Ovladač na tento paket neodpovídá, pouze se v horní části displeje zobrazí zpráva.

7.6 Vysílání s chybou v přenosu



Obrázek 7.7: Sejmutý průběh - Chybový přenos 1

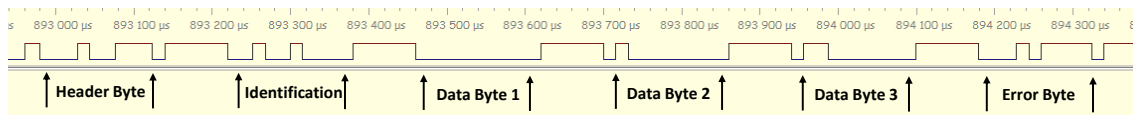
Na obrázku č. 7.7 je zobrazeno vysílání změny rychlosti a směru lokomotivy, které bylo popsáno v sekci č. 7.2. V kontrolním bytu, který má hodnotu 11110101, se nachází záměrná chyba. Tu rozpozná analyzátor a odešle paket o chybovém přenosu, který byl popsán v sekci č. 4.2.5. Ovladač musí na tento paket odpovědět potvrzením na reakci, které bylo popsáno



Obrázek 7.8: Sejmутý průběh - Chybový přenos 2

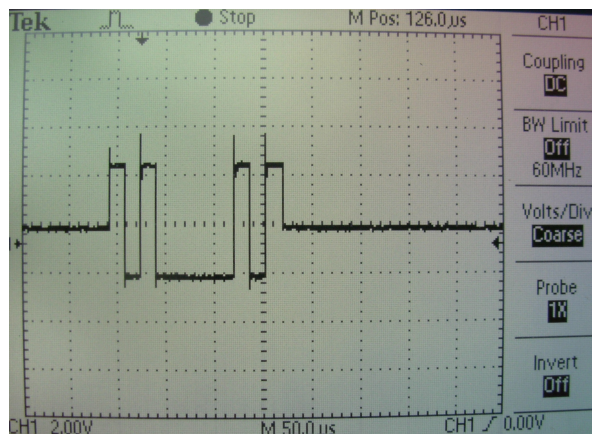
v sekci č. 4.1.3 Tento průběh je zobrazen na obrázku č. 7.8. Pokud by ovladač na paket o chybovém přenosu neodpověděl, příkazová stanice by ho dala do neaktivního stavu.

7.7 Komunikace reálného XpressNet systému



Obrázek 7.9: Sejmутý průběh - Změna rychlosti a směru lokomotivy

Na obrázku č. 7.9 je zobrazen průběh komunikace ovladače Lenz LH100. Jedná se o změnu rychlosti a směru lokomotivy. Rychlostní stupeň je 27, ovládaná je lokomotiva adresy 1, rychlost je 3 - dozadu. V porovnání s obrázkem č. 7.2 můžeme průběhy prohlásit za shodné.



Obrázek 7.10: Sejmутý průběh - Změna rychlosti a směru lokomotivy

Na obrázku č. 7.10 bylo sejmuto přenosové okno příkazové stanice pomocí osciloskopu. Je zde vidět, že komunikace probíhá na jiných napěťových hladinách než u komunikace přes UART. Napětí se pohybuje v rozsahu -2 V až 2 V a klidová hladina je 0 V. V případě komunikace vyrobeného ovladače s příkazovou stanicí je nutné použít převodník úrovně TTL a RS485. Datové vodiče z vývojové desky vedou zvlášť pro oba směry a na RS485 budou sloučené. Dále se zde bude nacházet řídicí bit pro přepínání - přijímač/vysílač.

Kapitola 8

Zhodnocení práce

Cílem práce bylo zhotovení ovladače, který bude schopen komunikovat po sběrnici XpressNet. V kapitole č. 1 a č. 2 jsem se seznámil s modelem kolejiště a také s dokumentací XpressNet, která byla při tvorbě ovladače nezbytná.

Dalším úkolem bakalářské práce bylo vybrat vhodný hardware, který byl popsán v kapitole č. 3 Hardware splňuje požadované funkce a umožňuje přidání dalších funkcí.

8.1 Ovladač lokomotivy

Zvolený hardware umožňuje široké možnosti realizace. Desku jsem rozšířil pouze o inkrementální spínač z důvodu pohodlnější regulace rychlosti a směru. Ovladač umí všechny základní funkce a o další funkce může být rozšířen.

8.2 Analyzátor XpressNet

Nad rámec bakalářské práce vznikl analyzátor, který dokáže simulovat důležité funkce příkazové stanice XpressNet. Toto zařízení by mohlo simulovat příkazovou stanici kompletně, ale pro vývoj ovladače to nebylo nutné. Při ladění ovladače byl tento analyzátor velmi nápomocný.

8.3 Plošný spoj

Protože ovladač byl doplněn o inkrementální spínač a při testování byl využíván analyzátor, tak byl vytvořen plošný spoj, který dané komponenty spojuje. Byl vytvořen z důvodu lepší prezentace výsledku práce a jeho ověřování.

8.4 Porovnání s reálným XpressNet systémem

V sekci č. 7.7 se porovnávala komunikace vyrobeným ovladačem s ovladačem Lenz LH100. Byl vyslán paket „Změna rychlosti a směru lokomotivy“ a komunikace byla na obou zařízeních

shodná. V případě testování vyrobeného ovladače s příkazovou stanicí XpressNet by bylo nutné doplnit ovladač převodníkem úrovní TTL a RS485.

8.5 Celkové zhodnocení

Ovladač lokomotivy funguje, tak je definováno zadání bakalářské práce a jeho funkčnost byla ověřena v kapitole č. 7. Řešení ovladače s konkrétním hardwarem umožňuje přidání nových funkcí a díky dotykovému displeji si grafické rozhraní můžeme přizpůsobit.

Literatura

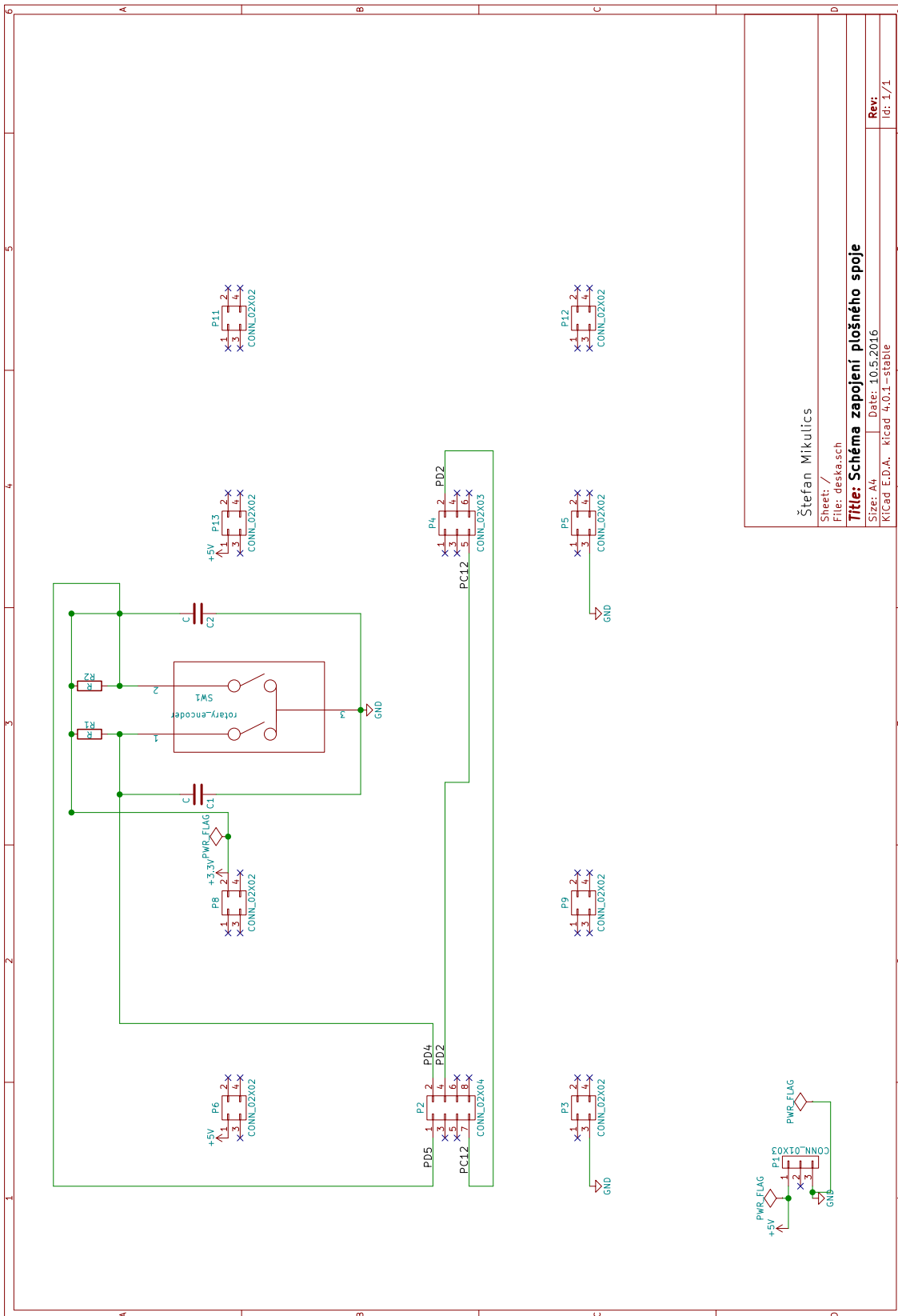
- [1] GM electronic, spol. s r. o. *Rotary Encoder RE24 - without switch* [online]. [cit. 16.4.2016]. Dostupné z: <<http://www.gme.cz/img/cache/doc/532/089/p-re24-datasheet-1.pdf>>.
- [2] KUŘÍK, O. *Studie řízení technologického systému, diplomová práce*. ČVUT v Praze , FEL, Praha, 2011.
- [3] Lenz electronic GMBH. *XpressNet Specification* [online]. Dostupné z: <<http://www.lenzusa.com/1newsite1/Manuals/xpressnet.pdf>>.
- [4] PERNER, J. *Řízení a vizualizace technologického systému, diplomová práce*. ČVUT v Praze , FEL, Praha, 2005.
- [5] Příspěvatelé Maxembedded.com. *Serial Communication - Introduction* [online]. 2013. [cit. 10.4.2016]. Dostupné z: <<http://maxembedded.com/2013/09/serial-communication-introduction/>>.
- [6] Příspěvatelé TechTerms.com. *Parity Bit* [online]. 2011. [cit. 16.6.2016]. Dostupné z: <http://techterms.com/definition/parity_bit>.
- [7] STMicroelectronics N.V. *Discovery kit with STM32F429ZI MCU* [online]. 2015. [cit. 16.4.2016]. Dostupné z: <http://www2.st.com/content/ccc/resource/technical/document/data_brief/ff/c1/b6/02/c3/b4/49/cb/DM00094498.pdf/files/DM00094498.pdf/jcr:content/translations/en.DM00094498.pdf>.
- [8] Tilen Majerle. *Library 26- Rotary encoder on STM32F4* [online]. 2014. [cit. 16.5.2016]. Dostupné z: <<http://stm32f4-discovery.net/2014/08/library-26-rotary-encoder-stm32f4/>>.

Příloha A

Schéma zapojení

Seznam součástek

Značka	Hodnota
C1	100nF
C2	100nF
R1	10K
R2	10K



Štefan Mikulics

Sheet: /
File: deska.sch

Title: Schéma zapojení plošného spoje

Size: A4 Date: 10.5.2016

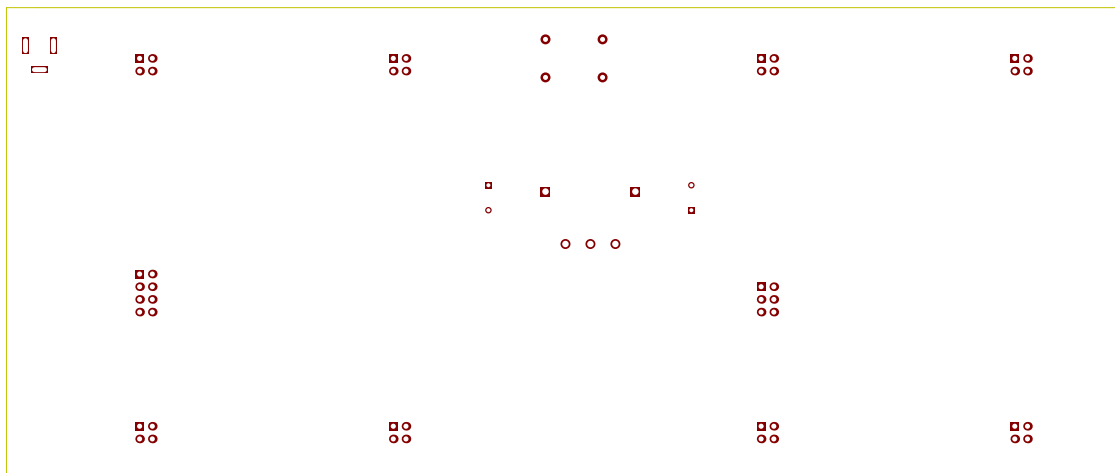
Rev:

Id: 1/1

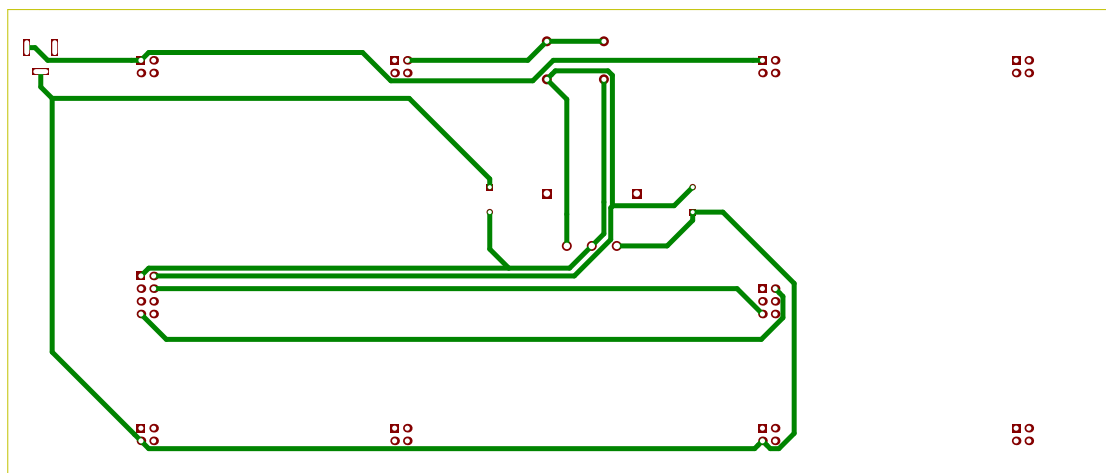
Příloha B

Deska plošného spoje

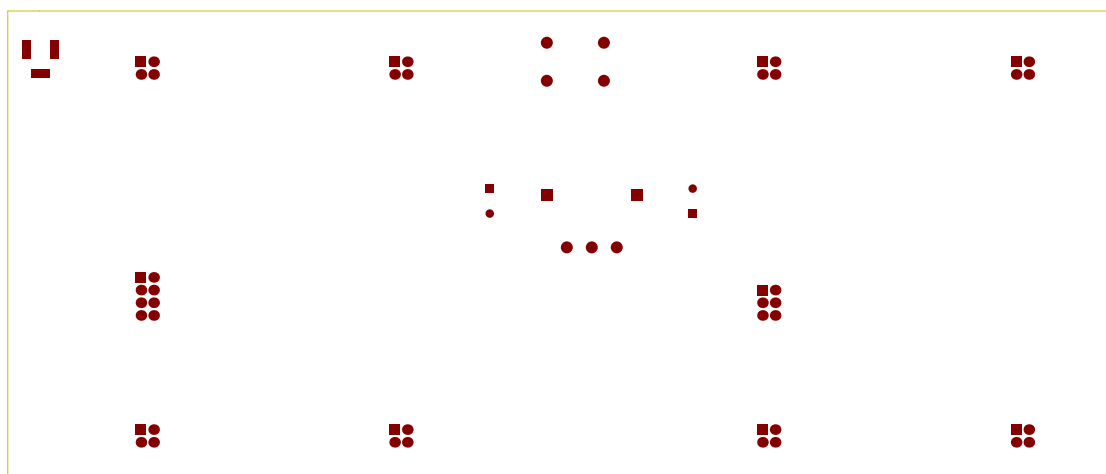
B.1 Přední strana - měď



B.2 Zadní strana - měď



B.3 Vrtý



Příloha C

Obsah příloženého CD

C.1 „Dokumentace“

Tato složka obsahuje dokumentaci k XpressNet protokolu a k vývojové desce STM32F429i.

C.2 „Plosny spoj“

V této složce se nachází KiCad projekt plošného spoje. Dále se zde nachází exportované schéma a výkresy.

C.3 „XpressNetAnalyser“

Zde se nachází μ Vision Keil projekt XpressNet analyzátoru včetně všech potřebných knihoven a souborů.

C.4 „XpressNetDriver“

V této složce se nachází μ Vision Keil projekt XpressNet ovladače včetně všech potřebných knihoven a souborů.